AL/HF-TR-1995-0187

ARMSTRONG LABORATORY

INTEGRATED MODEL DEVELOPMENT ENVIRONMENT (IMDE) SUPPORT FOR AIR FORCE LOGISTICS

Todd Carrico, Capt, USAF

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION
2698 G Street
Wright-Patterson Air Force Base, Ohio 45433-7604

Patrick K. Clark

TASC, INCORPORATED
2555 University Boulevard
Fairborn, Ohio 45324

July 1996

19961011 002

Final Technical Report for Period June 1993 to April 1995

Approved for public release; distribution is unlimited

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this report and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nations.

This report has been reviewed and is approved for publication.

Todd M. Carrico

TODD CARRICO, Capt, USAF
Project Director

BCWC

BERTRAM W. CREAM, GM-15, DAF
Chief, Logistics Research Division

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
July 1996

3. REPORT TYPE AND DATES COVERED
Final - June 93 to April 95

4. TITLE AND SUBTITLE
Integrated Model Development Environment (IMDE)
Support for Air Force Logistics

5. FUNDING NUMBERS
C - F33657-92-D-2055
PE - 62205F
PR - 1710
TA - D2
WU - 02

6. AUTHOR(S)
Todd Carrico, Capt, USAF
Patrick K. Clark

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TASC, Inc.
2555 University Boulevard
Fairborn, Ohio 45324

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
2698 G Street
Wright-Patterson AFB, OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL/HR-TR-1995-0187

11. SUPPLEMENTARY NOTES
Armstrong Laboratory Technical Monitor: Todd Carrico, Capt, USAF, AL/HRGO, DSN 785-2606

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This report documents research performed for the Integrated Model Development Environment (IMDE) by Armstrong Laboratory, Logistics Research Division, Wright-Patterson AFB, Ohio. IMDE is a simulation development system designed to demonstrate how an object-oriented modeling approach embedded within a graphical user interface could make large-scale logistics models easier to develop and less expensive to maintain, as well as improving configuration control, data analysis, collaborative development, and model reuse. The IMDE system was demonstrated through the development of an object-oriented fighter airbase logistics model. A significant achievement in the effort was the automated conversion of Logistics Composite Model (LCOM) into IMDE objects.

14. SUBJECT TERMS
Air Force Logistics
Discrete Event Simulation
Integrated Model Development Environment
Logistics Composite Model
Object Oriented Programming

15. NUMBER OF PAGES
55

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

*U.S.GPO:1993-0-358-779

CONTENTS

LIST OF FIGURES

PREFACE

This report documents work performed under the SIDAC contract (Contract No. F33657-92-D-2055), Task #28, entitled, "IMDE for Airbase Logistics". The Integrated Model Development Environment (IMDE) is a simulation development system developed under a contract performed by TASC and sponsored by the USAF Armstrong Laboratories Logistics Research Division. The original three-year effort was designed to support the objective of enhancing the Air Force's ability to perform simulation-based logistics capability assessments. The specific objective of the IMDE effort was to demonstrate how an object-oriented modeling approach embedded within a graphical user interface could make large-scale logistics models easier to develop and cheaper to maintain, as well as improving aspects of integrated configuration control, data analysis, collaborative development, and model reuse.

The IMDE effort successfully implemented an object-oriented system for developing, running, and analyzing simulations. IMDE consists of about 250,000 lines of government owned C++ code, a commercial object-oriented database management system, and a commercial simulation language compiler.

The IMDE system was demonstrated through the development of an object-oriented generic fighter airbase logistics model. Current USAF logistics modelers saw many potential advantages to using IMDE in the long run. However, in the short run there was the very significant hurdle of what to do about migrating existing legacy models. The Logistics COmposite Model (LCOM) is a standard USAF model used for manpower determination and spares provisioning, and is in widespread use at MAJCOMs and the Aeronautical Systems Center (ASC). LCOM has been successfully used for over twenty years, and as a result, databases exist for almost all USAF weapon systems, including special purpose systems. Users of LCOM are understandably reluctant to consider moving to a new modeling system, no matter what improvements it may offer, if they have to enter their existing databases manually into the new system. A major portion of the effort reported here was an attempt to investigate the feasibility of automatically converting existing USAF logistics data feeds into the IMDE object-

oriented format. This automatic conversion tool has been labeled the “flexible link” between the current forms-based system and the IMDE object-oriented system.

In addition to developing a "flexible link" to existing logistics data sources, this effort involved the comparison of the IMDE modeling capability to other systems currently in use, both through research into the existing models and through fielding IMDE at several sites and obtaining feedback on its advantages and disadvantages.

ACKNOWLEDGMENTS

The authors wish to thank the individuals listed below for their feedback and support during this task. Their contributions were essential to improvements to the products of this research and development project.

Capt. Pat Vincent, ASC/ALLH
Mr. Roger Gillilan, ASC/ALLH
Mr. Dick Cronk, ASC/XRE

The authors would also like to acknowledge the following former and current IMDE research team members.

Mr. Kerris Renken, TASC
Mr. Nick Stute, TASC
Mr. Jeff Sumner, TASC
Mr. Paul Barlow, TASC

The authors would also like to thank the countless individuals at the various MAJCOM's who provided guidance, ideas, feedback, and encouragement.

The authors sincerely appreciate the time and energy each of these individuals provided. Without their combined support, the IMDE project could not have succeeded.

SUMMARY

This report describes the results of research which extends work done on the development of the Integrated Model Development Environment (IMDE). IMDE is a set of software tools which together form a complete capability for the development of object-oriented discrete event simulations in a graphical environment. While the basic IMDE contract demonstrated the capability to develop complex models, potential users repeatedly stated that for any new modeling system to be accepted, it would have to support to a large degree the incorporation or conversion of the large number of existing legacy models. To date, we have demonstrated success in automatically converting a large number of the records contained in Logistics COmposite Model (LCOM) databases, without the need for user input. Here we describe our approach and implementation of the work done to date, as well as plans to convert the remainder of the database records.

A secondary task on this effort involved the evaluation of IMDE with respect to other modeling systems, such as LCOM, ISAAC, etc. We report our experience and thoughts about LCOM.

I. INTRODUCTION

The purpose of this report is to describe the results of Task #28 of the Supportability Investment Decision Analysis Center (SIDAC) contract. This effort, entitled "IMDE Support for Airbase Logistics", is sponsored by the USAF Armstrong Laboratories Logistics Research Division at Wright-Patterson AFB, OH. The work was performed by TASC, Inc. in Fairborn, OH. Previous joint work by Armstrong Labs and TASC resulted in the development of IMDE, a graphical, workstation-based system for the development, execution, and analysis of object-oriented simulation models. The primary goal of this task was the development of an automatic conversion program between existing sources of Air Force maintenance data and IMDE, so that IMDE will be better able to capture detailed information for simulating airbases, but without manual input of all of the data into IMDE's object database structure. Secondarily, this effort

looked at two other modeling systems which performed airbase simulations. One is LCOM, which has been used extensively for the last twenty years to assess manpower, spares, and support equipment requirements for new and upgraded aircraft weapon systems. The other is Integrated Simulation Assessment of Airbase Capability (ISAAC), an enhanced version of the Theater Simulation of Airbase Resources (TSAR), recoded in Ada with a menuing system for interfacing with the model base. ISAAC's development was sponsored by HQ USAF/LGS, but it was never operationally deployed for a variety of reasons, one predominant reason being the lack of integration with existing USAF maintenance data sources.

The scope of this effort has been limited to converting one type of Air Force data into IMDE format as a proof of principle. Since the data in different databases is similar, if one database can be successfully converted, it should be possible to convert all the other databases as well.

The structure of the report includes a background on the history of logistics modeling and previous IMDE work, followed by a detailed description of the design used to perform the conversion of LCOM databases to IMDE databases. Next, a brief comparison of features of LCOM, ISAAC, and IMDE is presented. The Future Work section describes potential enhancements to the current system, and finally, our conclusions about the current state of IMDE and its potential.

II. BACKGROUND

Air Force Logistics Modeling

The Air Force spends billions of dollars every year on support equipment, spare parts, and the manpower to keep planes flying. With modest investments in modeling and simulation tools, the Air Force has sought more efficient ways to acquire these costly items. The process of predicting requirements for airbase resources to sustain aircraft sorties during training and wartime is very complex. It requires not only a knowledge of the process of preparing aircraft to fly, but also the details of how long each step takes and what resources are required to perform

each step. These resources include manpower, parts, support equipment, and/or facilities. Detailed simulation has been recognized as a valuable tool for accurately representing the dynamic relationships on an airbase. Through simulation, a wide range of experiments with varied processes or asset levels can be quickly and accurately analyzed. The accuracy of the simulation's prediction is based heavily on the failure rates of different parts on the aircraft, mission profiles, aircraft configurations for different missions, manpower shift assignments, and other considerations. Some of this information, such as flying schedules, is currently gathered through interviews and conferences. However, a large amount of historical information is available from maintenance databases regarding the failure and repair of different subsystems aboard each weapon system. In particular, the Air Force collects a large amount of maintenance data by requiring that each maintenance action be documented with an AF Form 349, which specifies the activity being performed, on what part it is being accomplished, who is doing it, how much time is expended, and what resources were used. This raw data can be processed to create an average "logistics behavior" for each subsystem - how it fails and what it takes to be fixed. There are currently three sources of raw information that can be used to feed simulations of Air Force weapon systems: 1) The Maintenance Data Collection (MDC) system, 2) The Logistics Support Analysis Report (LSAR) system, and 3) The Reliability and Maintainability Information System (REMIS). The MDC system collects operational field data from Air Force units. LSAR data is the developing contractor's database, which includes laboratory test data and limited flight test data. The REMIS system is the next-generation "super"-MDC system currently nearing completion.

The Development of IMDE

The IMDE system is the result of almost 20 person years of effort to develop a comprehensive state-of-the-art simulation environment. Although tools like LCOM possess very powerful capabilities, they represent technology from the early 1970s. As a result, they require large amounts of tedious data preparation prior to simulation and a similar effort to convert output data into a spreadsheet-compatible format that can be easily graphed. Additionally, there is a long apprenticeship process to become a skilled LCOM user, the developed models are not easily reusable, and the visibility into what is actually happening in a model is not very clear to a non-"LCOMer." IMDE attempts to address many of these shortcomings. Internal configuration control of models and parts of models, object-oriented model development, graphical model descriptions, automatic input/output tracking, graphical data analysis, and a multi-user/multi-work group environment are some of the features IMDE has prototyped for a new generation of simulation developers and users. An overview of the IMDE environment is given in the final report to the initial contract, AL/HR-TP-94-0030. A comprehensive discussion of the IMDE software is provided in the User Manual.

Reception to the initial IMDE system from several modeling groups was very positive. Since numerous reviews and interviews with the modeling community were performed before designing the system, some degree of success was expected. However, one major obstacle to using IMDE or any other potential replacement to current environments was consistently raised during the three-year initial contract. Modelers demanded that a new system should have the ability to read in their existing archives of legacy models, so these considerable volumes of information would not have to be manually re-entered into modern systems. This problem was not trivial due to the complexity of some existing models and the fundamental difference between procedural and object-oriented methodologies. The laboratory felt that in order to obtain user buy-in and ensure success of the IMDE program, the capability to extend the system to include a conversion capability was essential. The term "flexible link" was coined as the data

conversion program that would be needed to take an existing Air Force logistics model as input, and create from it a set of IMDE model objects.

Developing a Flexible Link

Many programs exist that take raw data from maintenance information systems and generate a more composite picture of each of the subsystems on a weapon system. Two of these are of specific interest to our current work because they not only capture the time and resources involved in fixing each subsystem; they also capture the sequence of the repair process. These tools are the LCOM "front end" program maintained by HQ AFMEA and the MicroOmnivore program maintained by HQ AFOTEC/SAL. Both generate as their output an LCOM database, which can be used to directly simulate the detailed unscheduled maintenance activities for a particular weapon system. For this effort, we take advantage of these existing programs that digest the raw data by using their output, LCOM forms, as input for conversion to the IMDE object-oriented model. This provides two benefits. First, it keeps us from repeating work already done at least twice already, rolling up individual failures into composite data. Second, since a large portion of the simulation assessments currently done in the Air Force use existing LCOM forms databases, our flexible link will be able to convert existing simulation studies without having to revert back to the raw data, which may no longer be readily available for some systems. The converted models can be compared to the LCOM database counterparts directly to ensure compatibility. Figure 1 shows where in the data gathering and modeling process our flexible data link will do its work. Other models, such as TSAR, would have to have their own conversion written, since their input format is significantly different from LCOM. The scope of the current work focused only on the feasibility of automatic conversion of LCOM databases.

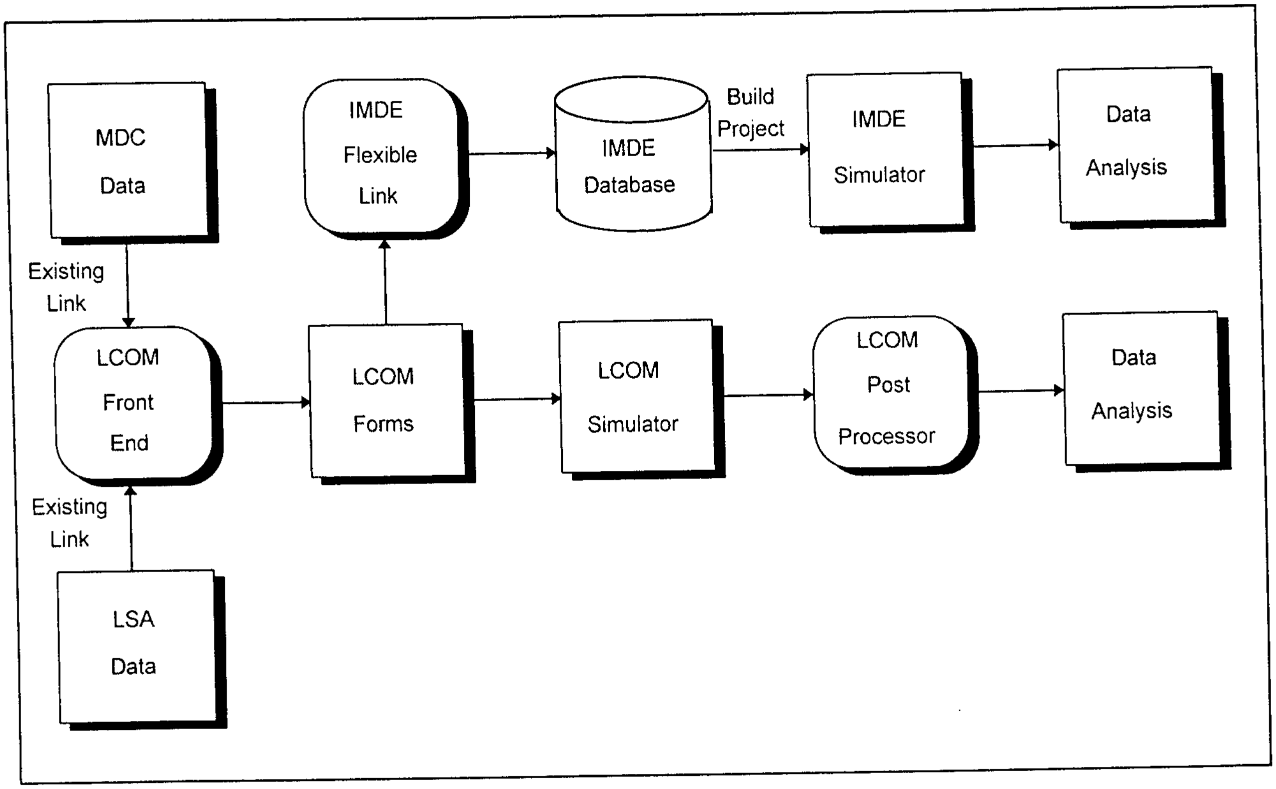


Figure 1.
Logistics Modeling and Flexible Link to IMDE

Making Comparisons

In addition to developing the flexible link to LCOM databases, this effort concentrated on trying to put IMDE into the hands of at least a few Air Force modelers currently using some other system, subsequently collecting feedback on which aspects of the new system were worthwhile, and identifying areas for improvement. During this task, we installed IMDE at HQ AFOTEC at Kirtland AFB, NM, at ASC/ALT at Wright-Patterson AFB, OH, and at HQ AFMEA, Randolph AFB, TX. Demonstrations and training were given to several groups during the course of the task, including analysts from HQ ACC, HQ AMC, ASC/XR, and ASC/ALT. Each of these groups felt there was significant potential for IMDE, but were unable to commit enough resources to make detailed recommendations. AFOTEC is currently in the process of using IMDE on a small "pathfinder" effort to develop a simulation of a Brilliant Eyes satellite constellation simulation model. Other development efforts are under consideration.

III. INTEGRATING EXISTING MODEL DATABASES

LCOM Forms

The task of converting existing model databases, simply stated, is to take the input form for one model and convert it into the input form for the IMDE system. We chose to demonstrate the conversion of the LCOM database format, which involves several different types of "forms," represented as 80 column records. These forms and their interaction are described in more detail in [Boyle,1991], and in much more detail in the LCOM User Manual [Cronk, 1990]. Before exploring the actual conversion process, a basic description of the primary LCOM forms will be presented.

There are 14 LCOM forms, each of which uses an 80-character record to specify its data. The data formats between the LCOM forms differ, so each is identified with a unique two-digit number in columns two and three of the record. The form numbers and titles are given in Figure 2. For the conversion effort performed to date, we have concentrated on Forms 15, 25, and 30. Together these forms account for 90% or more of all data records in typical LCOM databases. Plans for converting the other forms have been made, but their implementation is currently being prioritized by HQ AFMEA. During the initial conversion evaluation, these forms were input manually into IMDE.

Format Number	Form Title
10	Report Specifications
15	Resource Definitions
20	Attribute Definitions
25	Task Definitions
30	Task Networks
35	Clock Decrements
40	Distribution Definition
45	Shift Change Policies
50	Priority Specifications
55	Mission/Activity Definitions
60	Search Pattern Definitions
65	Internal Equipment Authorizations
70	Internal Equipment Group Definitions
75	Sortie Generation Data

Figure 2.
LCOM Forms

The Forms 15 in an LCOM database represent the pool of resources available for use by different tasks during the simulation, as well as the set of failure clocks used during the simulation. Resources can be specified as either men, parts, support equipment, or facilities. A seven-character name can be entered to identify the resource, as well as the quantity available, and substitute resources. Failure clocks typically specify mean values for failure of different subsystems. Actual values are drawn from the specified distribution each time the failure clock is "breached," which means it has decremented to zero from its original value. Figure 3 provides examples of both types of resource definition forms. Please note that the headings in Figure 3, as well as subsequent figures depicting LCOM data, are included for informational purposes only and are not part of the LCOM database.

Form Number	Resource Name	Type	Quantity	Failure Clock (First Parameter, Second Parameter, Distribution)		
15	FTRUCK	S	100			
15	D60	S	100			
15	ATE	S	100			
15	MJ1L0DR	S	100			
15	GUNL0DR	S	100			
15	L0XCART	S	100			
15	NF2LITE	S	100			
15	F110**	C		13.20	0.	X
15	F120**	C		23.77	0.	X
15	F130**	C		14.76	0.	X

Figure 3.
Example Forms 15

The Form 25 defines a task to be performed, giving its user-defined name, types and quantities of resources required to perform the task, and time duration parameters and distribution for the task. Figure 4 shows some examples of the Form 25. In looking at Figure 4, one can see that the resources specified for each task are resources that have been previously described on Forms 15. For instance, the first form in Figure 4 requires 1 unit of resource SHOP to perform a task, which has a mean duration of 2.0 hours, with a standard deviation of 0.58 hours. The distribution in this case is lognormal, given by the "L" code which follows the times. Other tasks in this example have different mean times for accomplishment, and require different quantities and types of resources.

Form Number	Task Name	Type	Priority	Task Duration (Mean, Std Dev)	1-3 Resources (Name, Consummable Flag 'C', Quantity)
25	W11***	7	3	2.00H .58HL	1SHOP 1
25	R11***	2	2	1.27H .37HL	1UMMT 2 SHELTER 1 NF2LITE 1
25	Q11***	2	2	0.64H 0.19HL	11*** C 1 1UMMT 2
25	N11***	2	3	1.00H .29HL	1SHOP 1
25	M11***	2	2	2.22H .64HL	1UMMT 2 SHELTER 1 NF2LITE 1
25	H11***	8	2	3.13H .91HL	1UMMT 2 SHELTER 1 NF2LITE 1
25	K11***	7	3	2.00H .58HL	1SHOP 1

Figure 4.
Example Forms 25

The Forms 30 represent the description of the sequence of steps performed during the modeled process. They specify which tasks (Forms 25) are performed in which order, with user-specified probability. A variety of selection modes allow exclusive (E) or alternative (nonexclusive) (A) branching, calling subnetworks (equivalent to subroutines) (C), or simple next node sequencing (D - Do mode). Figure 5 shows an example network section.

Form Number	Prior Node	Task Name	Next Node	Sel. Mode	Sel. Parameter	Comments
30	CALUM		UM0001	F	F110**	AIRFRAME
30	UM0001	R11***	R11001	E	.03000	R & R
30	UM0001	M11***		E	.96600	MINOR MAINT
30	UM0001	H11***		E	.00400	CND
30	R11001	SHOP	R11002	D		SHOP NETWORK
30	R11002	Q11***		I		CONSUME/CANNIB
30	R11002	G11***	I11001	D		RELESE ACFT
30	I11001	2LEVEL_MAINTENANCE	PDEPOT	E	.00000	ORG & DEPOT
30	I11001	3LEVEL_MAINTENANCE	I11002	E	1.0000	ORG,INT,DEPOT
30	I11002	W11***	PCYCLE	E	.33000	SHOP REPAIR
30	I11002	K11***	PCYCLE	E	.33000	RETEST OK
30	I11002	N11***	PDEPOT	E	.34000	SHOP NETWORK

Figure 5.
Example Forms 30

This extract of a network section from the simple LCOM "generic fighter" model is triggered by the decrementing to zero of the failure clock F110**, which occurs on the first line of the example. When the clock is breached, one of three possible paths is taken, indicated by the next three forms which have "E" selection modes and a probability of selection. In this example, 3% of the time a remove and replace action is required, which includes the task R11*** listed in Figure 4, but also includes additional processing at next node R11001. R11001 starts the series of events done at the repair shop once the subsystem has been removed (the fifth form in Figure 5). The path of the third form is taken 96.6% of the time, which represents a single task with no additional network. This single task is M11***, a minor maintenance task, again shown in Figure 4. The fourth form is similar, with a "could not duplicate" task being executed when this branch is taken. Figure 6 depicts the sequence of actions specified by the network in Figure 5.

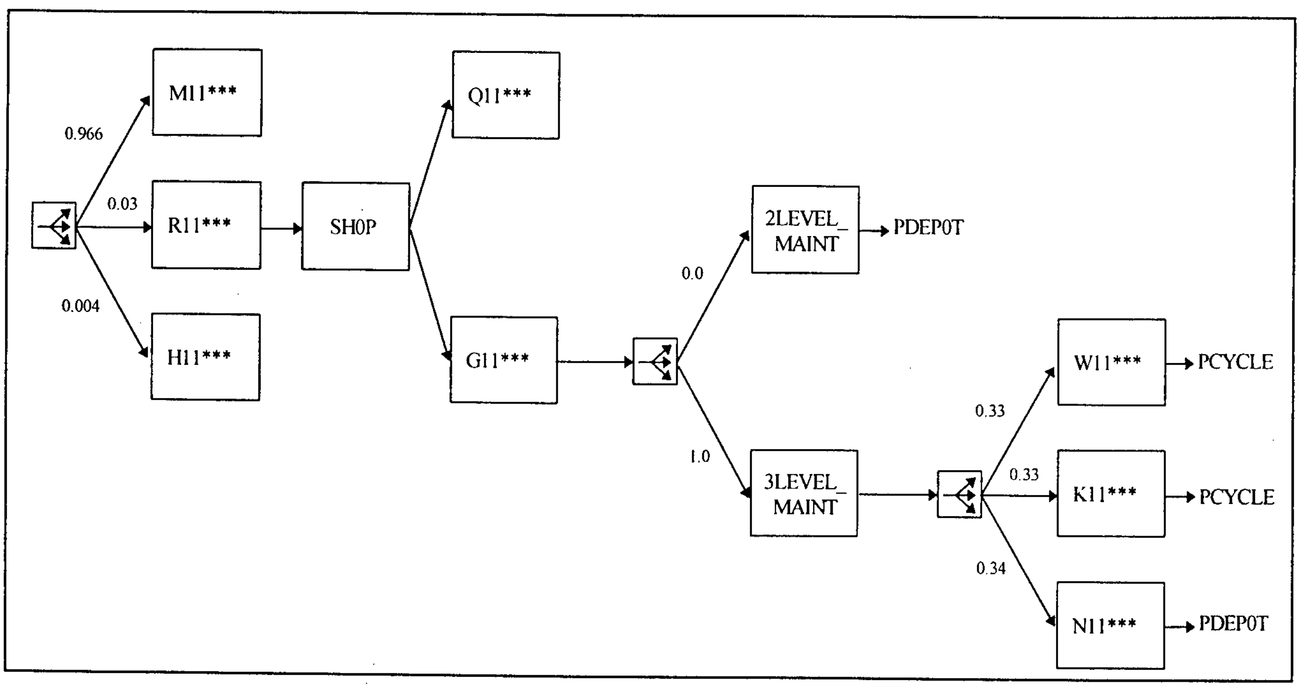


Figure 6.
Graphical Representation of F110 Network**

IMDE Objects

IMDE allows model construction to occur in a modular fashion, with parts of models being constructed which can potentially be reused in later simulations. These modular parts are called **object classes**. For example, a definition of an aircraft class as a model part would include the specification of the **attributes**, or variables, that describe the state of an aircraft during the simulation, and the **methods**, or functions, that describe the behavior of the aircraft and how it transitions between states during the simulation. Any number of object **instances** may be created from the class definition during the simulation. For example, a simulation may involve 48 aircraft, but they may all have been created using the same aircraft class definition, like a template.

Of considerable importance to IMDE, and in fact to the whole philosophy of object-oriented design and programming in general, is the concept of **inheritance**. Inheritance refers to the ability to reuse already-developed classes by extending them to have additional attributes and different or additional methods. A quick example involves considering a generic aircraft class with the basic attributes and methods required of all aircraft in the simulation study. Object-oriented programming languages allow us to create a more specialized class of aircraft, such as a fighter, as a child of the original aircraft class. When the aircraft class is specified as the parent of fighter, fighter automatically has all the functionality of aircraft, and the designer/programmer only has to specify different or additional features. This inheritance of characteristics of a more generic class is described in Figure 7, which shows the relation between aircraft and two children classes, fighter and tanker.

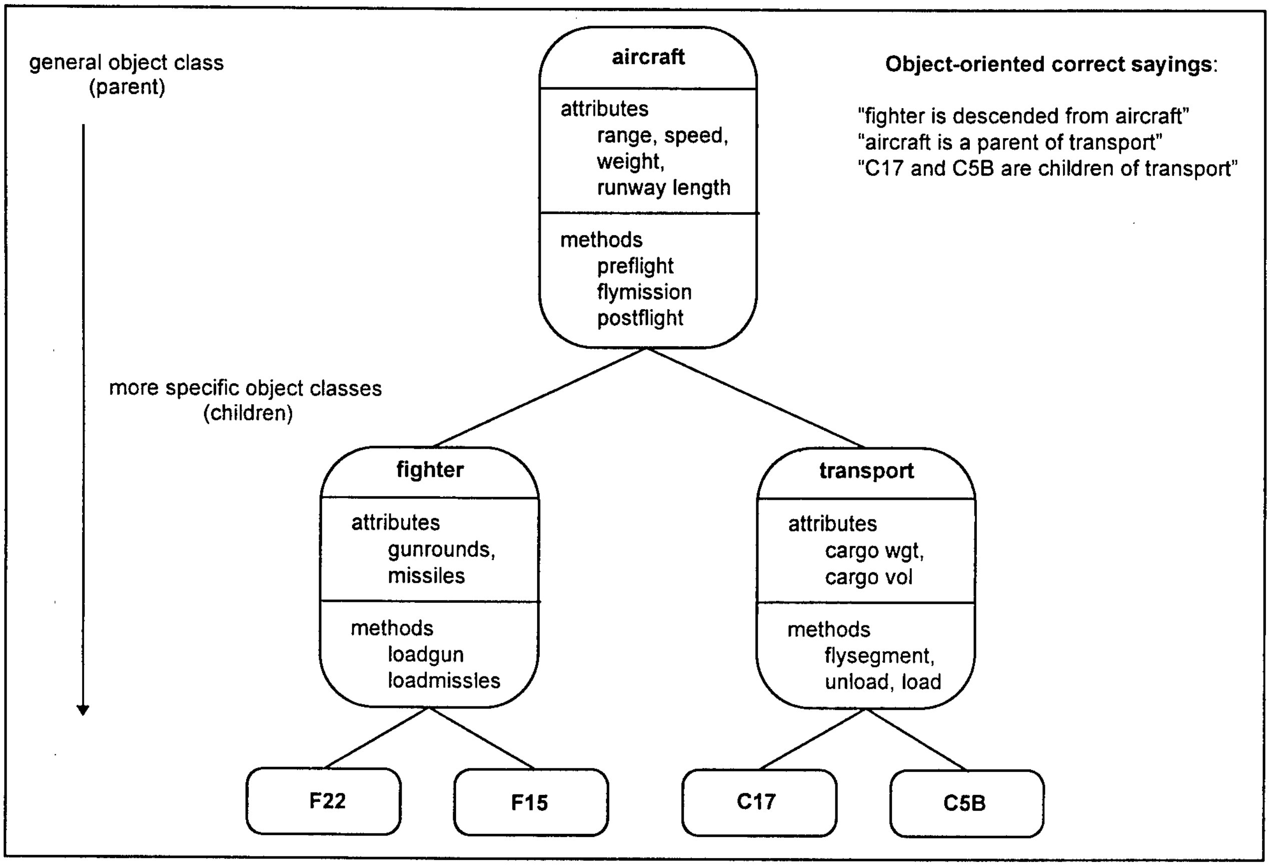


Figure 7.
Object-Oriented Inheritance

Further specialization of classes is possible by creating children of fighter and tanker, as also shown in Figure 7. During the course of the original IMDE contract, a set of airbase logistics model classes was developed in the process of creating a relatively small (30,000 lines of code) simulation program. This object set consisted of "concrete," real-world objects like aircraft, parts, test equipment, and people of different skills, as well as organizational objects like aircraft maintenance units, shops, squadrons, and theater. It also included some abstract classes such as a mission type object, scenario object, mission generator, and reconfiguration object. Since our example simulation was patterned after the basic LCOM generic fighter model, many of these objects will be in large part reusable for the more complex simulations involving real world aircraft system models. The different LCOM forms will map to different objects in the generic fighter model set, using them as parent classes from which the specific data represented

in the forms will be added to create child classes. For example, the sortie generation forms (Forms 75), can be mapped directly into the generic fighter model MissionTypeObj class without adding any new attributes or methods, although new classes will still be created to set appropriate default values for the attributes and override the default reconfiguration methods. Children of MissionTypeObj for the test F16 database from HQ ACC would include a CombatAirPatrolObj, CloseAirSupportObj, and EscortObj. Each of these would have different values for the parameters shown in the IMDE specification of MissionTypeObj (Figures 8 and 9). Parameters like mission duration, frequency, launch configuration, day mission percentage, etc., would be set differently for the different actual mission types. The other forms correspond in a similar fashion to objects in the generic fighter model. The search patterns (Forms 60) are very close in structure to the specification for the IMDE ReconfigObj (Figure 10). Some of the forms may need to be integrated into more than one IMDE object. The shift change policies for manpower resources (Form 45) is one possible example. The majority of form records in a database fall into the resource, task, and network definition forms. For this reason, our efforts for this project have targeted the conversion of these forms.

Class Name: **MissionTypeObj**

Group: fx99

Description: created from scratch

Parents:

Premod

Children:

EscortMissionType

CombatAirPatrol

Attributes:

NAME	*TYPE*	*DEFAULT*	*AUTO*	*STATS*	*PUB*	*CLASS*	*LIST*
Configuration	ReconfigObj		X		X		
Cycle	INTEGER	1	X				
DayMissionCompleted	INTEGER	0		X	X		
DayMissionPercentage	REAL	0.67	X				
DayMissionTotal	INTEGER	0		X	X		
FltLeader	AircraftObj				X		
FlyingHours	REAL	0.0			X		
LaunchConfiguration	STRING	MISSLS	X		X		
LeadTime	REAL	3.0	X		X		
MaxAircraft	INTEGER	2	X		X		
MinAircraft	INTEGER	1	X		X		
MissionDist	STRING	LogNormal			X		
MissionLengthParm1	REAL	2.2	X		X		
MissionLengthParm2	REAL	0.44	X		X		
MissionType	STRING	CAP	X		X		
MissionsCompleted	INTEGER	0		X	X		
PostSortieTime	REAL	0.0		X	X		
PreSortieTime	REAL	0.0		X	X		
SortieDuration	REAL	0.0		X	X		
TakeoffDist	STRING	Uniform	X		X		
TakeoffDistParm1	REAL	12.0			X		
TakeoffDistParm2	REAL	0.0			X		
TakeoffTime	REAL	24.0	X		X		
TotalMissions	INTEGER	0		X	X		
TotalSorties	INTEGER	0		X	X		

Figure 8.
IMDE Specification for MissionTypeObj (Attributes)

Class Name: **MissionTypeObj**

Group: fx99

Description: created from scratch

Parents:

Premod

Children:

EscortMissionType

CombatAirPatrol

Methods:

NAME	*PARAMETERS*	*TYPE*	*PUBLIC*	*CLASS*	*OVERRIDE*
AddDayMissionComplete	INTEGER	ASK	X		
	MissionObj				
AddMission	INTEGER	ASK	X		
	INTEGER				
AddMissionAbort	INTEGER	ASK	X		
	INTEGER				
AddMissionComplete	INTEGER	ASK	X		
	MissionObj				
AddSortiesComplete	INTEGER	ASK	X		
EnterPostSortieTime	REAL	ASK	X		
EnterPreSortieTime	REAL	ASK	X		
EnterSortieTime	REAL	ASK	X		
GenDailyMissions		TELL	X		
Init		ASK	X		
PrepToFly	AircraftObj	TELL	X		
	IMDETrigger				
PrepToPostflight	AircraftObj	WAITFOR	X		

Figure 9.
IMDE Specification for MissionTypeObj (Methods)

Class Name: **ReconfigObj**

Group: fx99

Description: created from scratch

Parents:

Premod

Children:

SP1MISSLS

Keywords:

Attributes:

NAME	*TYPE*	*DEFAULT*	*AUTO*	*STATS*	*PUB*	*CLASS*	*LIST*
CutoffTime1	REAL	0.0	X		X		
CutoffTime2	REAL	2.2	X		X		
CutoffTime3	STRING	2.7	X		X		
DesiredConfig	STRING	MISSLS	X		X		
FromConfig1	STRING	MISSLS	X		X		
FromConfig2	STRING	CLEANM	X		X		
FromConfig3	STRING	CLEANB	X		X		
FromConfig4	STRING	BOMBS	X		X		

Methods:

NAME	*PARAMETERS*	*TYPE*	*PUBLIC*	*CLASS*	*OVERRIDE*
Reconfig1	AircraftObj	TELL	X		
Reconfig2	AircraftObj	TELL	X		
Reconfig3	AircraftObj	TELL	X		
Reconfig4	AircraftObj	TELL	X		

Figure 10.
IMDE Specification for ReconfigObj

The Conversion from LCOM to IMDE

Capturing the information in the three most often-occurring forms, will require integration into more than a single IMDE generic fighter class. To some extent, this reflects choices made in the original generic fighter simulation. Other designs may have been able to isolate the effects of each form more singularly, but we feel this would have been at the cost of making the model structure less reusable for potential non-LCOM applications.

To figure out how to systematically convert these three forms, we were fortunate to have the expert assistance of several LCOMers, first to give us a primer in LCOM modeling, and

second, to provide us with several example databases. The most representative of a complete weapon system model was the ACC F16 database. While discussions indicated that this should be a good candidate to use as a conversion template, we hope to be able to eventually acquire and convert several others to quantify the generality of our conversion methodology.

The first step was to read in a complete LCOM database. We generated C++ data structures for each of the forms in order to have faster access to each field of the form. We built a forms viewer to verify that the forms were being read correctly into these structures. The forms viewer was modified to allow editing forms before converting a database into IMDE format. This forms editor may be somewhat useful by itself to current LCOM users. LCOM is being moved to a Unix workstation as the standard platform, which doesn't have the column-sensitive editors users are accustomed to on the IBM mainframe. Since the forms editor contains the specific LCOM forms, it could serve as a good replacement, allowing users to enter forms without having to count columns to determine field start and end points. The screens for the forms viewer/editor are shown in Appendix A.

After successfully reading the forms into a C++ format, we started to examine the Forms 30 in the F16 database. There are three major sections of these forms: 1) the main line networks, 2) the reconfiguration networks, and 3) the unscheduled maintenance networks. The main line networks represent the aircraft/mission flow process for each kind of mission. Parts of these have been integrated into the IMDE MissionTypeObj previously discussed. The reconfiguration networks will be integrated into IMDE ReconfigObj. These two network sections have not been converted as of this writing, but their implementation should be fairly straightforward, given the success we have had with the unscheduled maintenance (UM) networks. The UM networks represent the flight line, shop, and depot maintenance actions generated as a result of subsystem and component level failures during a sortie. They represent about 60% of the 10,120 records in the F16 database, so implementing them first had high priority.

The UM network section is naturally divided into many subsections of Forms 30 records, each representing a subsystem on the F16 that may fail during the simulation. This network section specifies how often a failure of any kind occurs for the subsystem through the failure

clock that begins the section, the sequence of tasks to be performed in the maintenance of this subsystem (some tasks possibly performed in parallel), and the resources and times to perform those tasks. This LCOM flow specification follows the previous description given for the interaction of Forms 30 with task and resource definition forms. Within IMDE, it was decided that each of these network sections would be associated with an IMDE object representing that part in the database. The generic fighter model includes an IMDE class called PartObj, which is a generic model of a component part or subsystem of a higher level system (which could as easily be a tank or a ship as an aircraft). PartObj includes attributes such as mean time to repair (MTTR), mean time between failures (MTBF) (for storage of the LCOM failure clock), and a variety of others. The IMDE specification for PartObj is provided in Figure 11. For each network section in the LCOM database, a child class of PartObj was created having the name of the work unit code of that LCOM section. For example, if a network section started off with checking failure clock F11*** (airframe subsystem), then a class called wuc11XXX was generated in the IMDE database. Its value for time between repairs would be set to the failure clock value. Every PartObj and descendent of PartObj has a method called FixPart, which describes how that part is repaired given that its failure clock decrements to zero. Each subsystem will in general have a different set of steps within its FixPart method. These steps are automatically generated by reading the LCOM networks for the subsystem and creating an equivalent set of IMDE graphical methods, all without user interaction. Once the LCOM networks have been read into the IMDE PartObj's FixPart method and sub-methods, the IMDE user can easily change the methods graphically to reflect an actual or proposed process change and evaluate the effects. Currently, the automatic creation of networks, assignment of probabilities to IMDE branching nodes, and completion of task parameters (required resources and time delays) have all been demonstrated for UM networks. Figure 12 shows a subsection of the F16 database representing the Forms 30 for one particular aircraft subsystem, and Figure 13 shows a graphical representation of this process. IMDE translates this process flow intographical diagrams called networks. Each of the "nodes" in the network drawings represents a flow or simulation construct used to graphically define an algorithm as a chronological sequence of events. Figure 14 shows the palette used to select nodes, which comprises the IMDE networks.

Class Name: **PartObj**

Group: fx99

Description: auto created

Parents:

Children:

- wuc11
- ECMPodObj
- Radar
- Antenna
- RSP

Keywords:

Attributes:

NAME	*TYPE*	*DEFAULT*	*AUTO*	*STATS*	*PUB*	*CLASS*	*LIST*
Aircraft	AircraftObj				X		
CountDownDist	STRING	Exponential	X		X		
FailCountDown	REAL	5.0	X		X		
FailCountDownParm1	REAL	5.0	X		X		
FailCountDownParm2	REAL	0.0	X		X		
Fixed	BOOLEAN						
Location	AirbaseObj				X		
MTBF	REAL	5.0	X		X		
MTBM	REAL	3.0	X		X		
MTTR	REAL	2.0	X		X		
MTTRDistType	STRING	LogNormal	X		X		
MTTRParm2	REAL	0.58	X		X		
NxtLowerAssemblies	PartObj	0	X		X		X
OnAircraft	BOOLEAN				X		
OrigEquipment	BOOLEAN				X		
TaskList	TaskObj	0	X		X		X
TwoLevelMaint	BOOLEAN				X		
rollupflag	BOOLEAN				X		

Methods:

NAME	*PARAMETERS*	*TYPE*	*PUBLIC*	*CLASS*	*OVERRIDE*
AssessDamage	AircraftObj	ASK	X		
CheckDamage	IMDETrigger	TELL	X		
FixPart		TELL	X		

Figure 11
IMDE Specification for PartObj

Form Number	Resource Name	Type	Quantity	Failure Clock	
15	F11E**	C		56.00	0.

Form Number	Task Name	Type	Priority	Task Duration (Mean, Std Dev)	Distribution	1-3 Resources (Name, Consummable Flag 'C', Quantity)	
25	V11E00	2	1	3.500H(1.015H)	L	454M0	1
25	M11E00	2	1	3.500H(1.015H)	L	454M0	1
25	M11E01	2	1	3.500H(1.015H)	L	454M0	1
25	R11E00	2	1	3.500H(1.015H)	L	454M0	1
25	DNRTFL	2	1	3.500H(1.015H)	L	454M0	1
25	Q11EFL	2	1	3.500H(1.015H)	L	454M0	1
25	G11EFL	2	1	3.500H(1.015H)	L	454M0	1
25	N11EFL	2	1	3.500H(1.015H)	L	454M0	1
25	PDEPOT	2	1	3.500H(1.015H)	L	454M0	1

Form Number	Prior Node	Task Name	Next Node	Sel. Mode	Sel. Parameter	JCN Count	Comments
30	CALLS1		A1E01	F	F11E**	11E**	0 CENTE
30	A1E01	V11E00		A	.078	11E**	0 CENTE
30	A1E01	M11E00		E	.629	11E**	0 CENTE
30	A1E01	M11E01		E	.185	11E**	0 CENTE
30	A1E01	R11E00	IA1E00	E	.186	11E**	0 CENTE
30	IA1E00	DNRTFL	IA1E01	D		11E**	0 CENTE
30	IA1E01	Q11EFL		D		11E**	0 FLIGH
30	IA1E01	G11EFL	IA1E02	D		11E**	0 FLIGH
30	IA1E02	N11EFL	PDEPOT	D		11E**	0 FLIGH
30	PDEPOT	PDEPOT		D			

Figure 12.
Subsection of F-16 LCOM Database

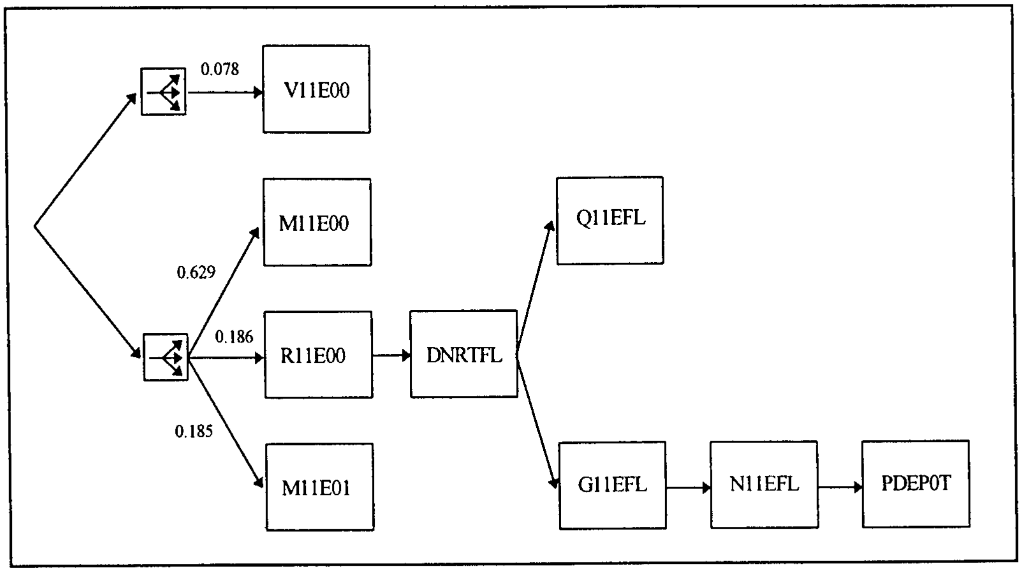


Figure 13.
Graphical Representation of F11E Network**

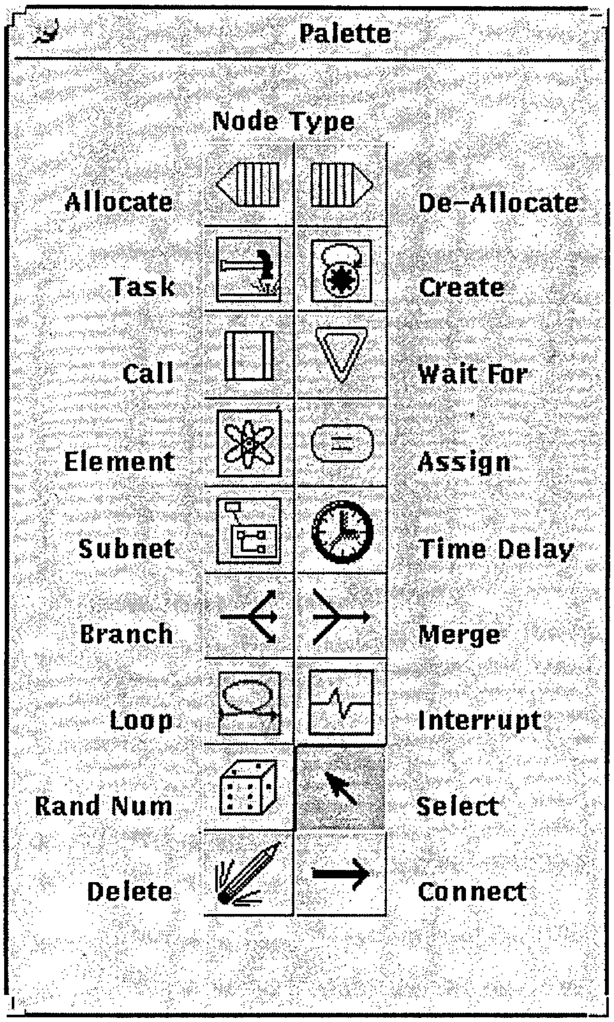


Figure 14.
Network Editor Palette

The process of reading and converting the LCOM F16 database originally took about four hours. The algorithm performed several exhaustive n^2 searches, which have since been eliminated, improving the process to about a 40-minute run time. This dramatic improvement should make rapid conversions possible.

IV. COMPARING IMDE TO OTHER LOGISTICS SIMULATION SYSTEMS

In addition to the development of the flexible link to data systems, this task order involves the analysis of other USAF logistics models and modeling systems to determine the relative advantages and disadvantages of IMDE. ISAAC will be evaluated in the next half of this task. In the process of developing the flexible link to LCOM forms, we have acquired a fundamental knowledge of LCOM capabilities, and present an initial comparison with IMDE in this section.

LCOM was designed in the late 1960s using the Simscript discrete-event simulation language. To run an LCOM program, a standard compiled simulation engine is used to read a file of formatted data that captures the specific information for the system under study. This file consists of the forms discussed previously in this report (complete list given in Figure 2). The different set of forms essentially comprises a limited simulation language, with column sensitive restrictions similar to those found in punch-card era languages. Unfortunately, this is not a standard programming language, so analysts have to be specifically trained to use LCOM. This training is usually available only "on-the-job," and takes one or two years to acquire at a fully qualified level. Once trained, users of LCOM have at their disposal a powerful tool for simulation. Changing manpower levels, parts quantities and reliabilities, aircraft mission profiles, and even the event sequencing of different airbase processes allows users to make a wide range of studies to explore potential effectiveness and efficiency improvements to airbase operations. Over the last 20 years, LCOM databases have been developed for almost every aircraft weapon system, and many of these databases are still in use. LCOM is the standard Air Force system for assessing manpower requirements, and is also used extensively for spare parts provisioning studies.

While LCOM is unquestionably the system of choice used by today's manpower analysts, it faces some challenges in the future. These challenges include a limited capability to model certain processes, the importance of simulation to the Department of Defense (DoD), and the feasibility of training new personnel on outdated technology.

The LCOM simulation engine is limited in its capability to represent some potentially important processes, such as multiple leg mission simulations required by Air Mobility Command (AMC). Although LCOM is in fact currently used to perform these simulations, there is general dissatisfaction with the "gaming" of LCOM that is necessary. There are several other known limitations inherent with the simulation engine. While these limitations could probably all be lessened or eliminated by changes to the simulation engine, the "upgraded" simulation engine would be as much of a "black box" as the original.

Another challenge lies in the fact that simulation has become an important DoD "thrust area". The DoD plans to use simulation much more in the future to lower operations and maintenance costs by more accurately predicting cost drivers. When budgets were large, it was much easier to buy more equipment and manpower in many different areas to ensure that there was never a shortage. If decision makers are going to use simulations to guide where they allocate their more limited funding, they will rightly insist on having more insight into how the simulation model is designed than is currently available with LCOM. People want to see how the processes are modeled in a graphical representation and be able to change those processes easily as operational considerations change. The models of the future will have to be transparent models, where each user or user group can see exactly what the model is doing at any level of detail, without having to look at a programming language level description. These transparent models will allow the model developer to communicate to the decision maker why the model correctly represents the system about which an important decision will be based.

A third challenge for LCOM is to attract new modelers to be trained in the 20-year-old technology. While the LCOM forms style of building models was truly advanced when introduced, it is now very outdated and cumbersome compared to modern graphical user interface technology. Many existing LCOM users were trained when there was no other option. Those users will eventually retire or move on to other, non-programming positions. The increased emphasis on the importance of simulation, combined with the precipitous manpower drawdown, makes it imperative to attract a new generation of bright, motivated modelers, and to give them the tools to help them perform their job more effectively. In the current environment, people are looking harder than ever at keeping current at skills that will be fungible in the

commercial job market. LCOM will not be viewed by many systems engineers, operations research analysts, or computer scientists as a tool to keep them up to date on technology, and they will consequently tend to look for other opportunities in other jobs. Given the training required to learn LCOM, the scenario of modelers potentially leaving an LCOM shop after a few years would make it difficult to maintain a capable organization.

IMDE has the potential to address the needs of the LCOM modeling community, bring greater involvement and understanding to decision makers, and bring newer technology to modelers. With the completion of this task, a comprehensive set of airbase objects compatible with an LCOM database will have been designed within IMDE. These objects will serve as the start of an IMDE database, a majority of which can be created automatically from an existing LCOM database. Potentially, most, if not all, existing LCOM databases could be converted with only a small manual effort, giving equivalent modeling capability to IMDE. Since this effort does not actually validate the IMDE outputs with LCOM equivalents, some analysis would be necessary to explain differences. If significant differences exist, either IMDE objects would need to be modified, or the differences would need to be justified based on inadequate or incorrect design of LCOM.

Since IMDE allows the graphical creation and editing of process descriptions, decision makers can look inside a model to get insight into the process steps considered, without having to read SIMSCRIPT, LCOM, or some other programming language. If the process needs to be changed, steps can be added interactively in the process description, right where the decision maker wants them. This view also will help the modeler, allowing him to see as much of the simulation engine as he needs, and possibly tailor it to extend the functionality to fit his needs.

Incorporating object-oriented design, programming, and databases with an X-window interface into IMDE makes it a tool that can provide the next generation simulation developer with an exposure to the latest workstation modeling capabilities. IMDE is a domain-independent set of tools with potential applications outside airbase logistics modeling, which provides a methodology for designing simulation models applicable to many problem areas, including commercial sector applications. In addition to keeping the modeler in tune with current technology, IMDE provides the obvious benefits of built-in model configuration control and data

analysis, very important capabilities which were often neglected or second-rate in the last generation of simulation tools.

IMDE Transition

There are several potential challenges in transitioning IMDE for use by the Air Force. These include training, support, cost, validation, and conversion of legacy databases.

IMDE is a complex tool that will require significant training for the average Air Force analyst. An aggressive training program must be in place to assure IMDE's success with the modeling community. IMDE currently includes a significant on-line help capability, but an on-line tutorial should be added to provide a total walk-through of simulation project development.

The support of any complex tool is very important. Just as LCOM currently is supported by at least six full-time programmers/analysts, IMDE would require some cadre to maintain and enhance the software. Support would range from answering user phone inquiries, to minor bug fixes, to potentially large enhancements, such as an additional user level, with parameterization greatly simplified over the existing IMDE Experiment Editor. With current specially-designed tools like LCOM, the total support cost must be borne by the government, since no one else is using them. With IMDE, this support could potentially be spread across commercial users, since IMDE is equally well suited to developing manufacturing models. In order for this to happen, IMDE must be made into a commercial product.

Another potential obstacle to transitioning IMDE to general use is the current cost of a complete IMDE workstation. IMDE requires a Sun workstation, currently a minimum of \$5,000, as well as commercial software costing approximately \$24,000. Efforts are underway to reduce the software costs, either by bundling agreements with current vendors, finding alternate commercial sources, or developing government-owned equivalents. Hardware costs could be reduced by porting IMDE to a PC platform, either to the Solaris™ Sun operating system, which would be a fairly small development, or to Windows NT™, which would involve a significant development effort.

As previously mentioned, acceptance of IMDE-developed models will hinge on validation of those models, first against LCOM equivalents. If the two models match fairly well, the validation effort might be fairly small, limited to output data analysis. If there are significant differences in critical measures of merit in the output, validation will require evaluation of which system, LCOM or the IMDE model, is correctly modeling the real-world situation. This could be a significant effort, especially if the difference is due to a suspected flaw in the LCOM engine, since the "black box" of LCOM would have to be dissected and understood.

Finally, the key to eventual transition to IMDE will be the conversion of existing large databases into IMDE format. For the LCOM community, a large step will have been taken to implement this conversion by the end of the current effort. For other models, the conversion effort may be similar in scope if a standard set of inputs exists and is well-documented.

V. FUTURE WORK

The majority of effort to date has been spent developing a detailed conversion methodology for LCOM forms. The first proof of this methodology is well under development with the conversion of the unscheduled maintenance portion of the Forms 30 and associated Forms 15 and 25. Now that the unscheduled maintenance section has been fully converted, the remaining major Forms 30 sections, which represent the main mission networks and reconfiguration sections can be converted as well. LCOM attribute definitions (Forms 20) will probably have to be implemented as well, most likely as attributes of IMDE object-oriented classes. Finally, conversion of the LCOM change card file, which provides for a smaller set of parameter changes, will be implemented. Together, these pieces of the overall effort should result in a system which will convert nearly 90% of existing LCOM databases.

VI. CONCLUSIONS

From a technology standpoint, this effort has already demonstrated the feasibility of converting a record-oriented legacy database into a more modular, reusable, and maintainable object database format. There are clear advantages in transitioning to a more modern modeling system, with a more graphical interface and less programming language orientation. The resistance to doing so will be no different than is inevitably found in other automation systems that have been in service for several years: people like what they are used to using. The key to overcoming this resistance is to demonstrate clearly that the new system provides a superset of the capabilities of the old system, and does so more effectively and efficiently. In the case of IMDE, this demonstration should be greatly facilitated by the development of the data conversion program from existing LCOM forms. For full acceptance, validated studies will probably have to be done comparing the results of models converted to the IMDE system with the original LCOM results. Differences in the output will be assumed to show IMDE as incorrect, due to LCOM's status as a standard system. It may take a significant effort to demonstrate that differences may be due to limitations in the LCOM model, as opposed to problems in the IMDE-converted model. Even successful validation will not be adequate to guarantee IMDE's success as a follow-on system for airbase logistics modeling. The tools and training to deploy IMDE with several modeling groups, perhaps initially on small pilot projects, will help to develop a corps of "insiders" who believe in the advantages that IMDE can offer, and who will be more convincing in talking to others in their organizations about the future tools of choice.

GLOSSARY

attributes - variables in an object class that describe the state of an instance of that class at any point in the simulation. For example, an F-16 class may have attributes such as MaxSpeed, Heading, FuelAmount, Weapons, etc.

flexible link - the data conversion program that accepts an existing Air Force logistics model as input and creates a set of IMDE objects.

inheritance - The ability to reuse previously developed classes by extending them to have additional attributes and different or additional methods. For example, an F-16 class may inherit attributes and methods from a Fighter class, which may in turn inherit attributes and methods from an Aircraft class.

instances - Specific occurrences of an object class in a simulation. For example, a simulation that models eight F-16s will have a single F-16 class, but the simulation will have eight instances of the class.

methods - functions of an object class that describe the behavior of the class as well as how instances of that class will change states during the simulation. For example, an F-16 class may have methods such as FlyMission, FireMissile, Refuel, etc.

object classes - representations of real-world entities used to model these entities in an object-oriented environment. Object classes are reusable, and are comprised of attributes and methods.

LIST OF ACRONYMS

IMDE	Integrated Model Development Environment
ISAAC	Integrated Simulation Assessment of Airbase Capability
LCOM	Logistics COmposite Model
LSAR	Logistics Support Analysis Report
MDC	Maintenance Data Collection
MTBF	Mean Time Before Failure
MTTR	Mean Time To Repair
REMIS	REliability and Maintainability Information System
SIDAC	Supportability Investment Decision Analysis Center
TSAR	Theater Simulation of Airbase Resources
UM	Unscheduled Maintenance

REFERENCES

Boyle, Edward (1991, July). *LCOM Explained* (AFHRL-TP-90-58, AD-A224497). Wright-Patterson AFB, OH: Logistics Research Division, Armstrong Laboratory.

Clark, Patrick, and Lloyd, Bradley (1994, February). *Object-oriented Simulation in IMDE.* Proceedings of the Western Simulation Multi-Conference 1994, Object-Oriented Simulation Conference.

Cronk, Richard, and Wallace, Alan (1992). *LCOM User Manual Version 93.C.* Wright-Patterson AFB, OH.

Lloyd, Bradley, and Clark, Patrick (1993, November). *The Integrated Model Development Environment* (AL-TP-1992-0030). Wright-Patterson AFB, OH: Logistics Research Division, Armstrong Laboratory.

Appendix A

LCOM Forms Viewer/Editor

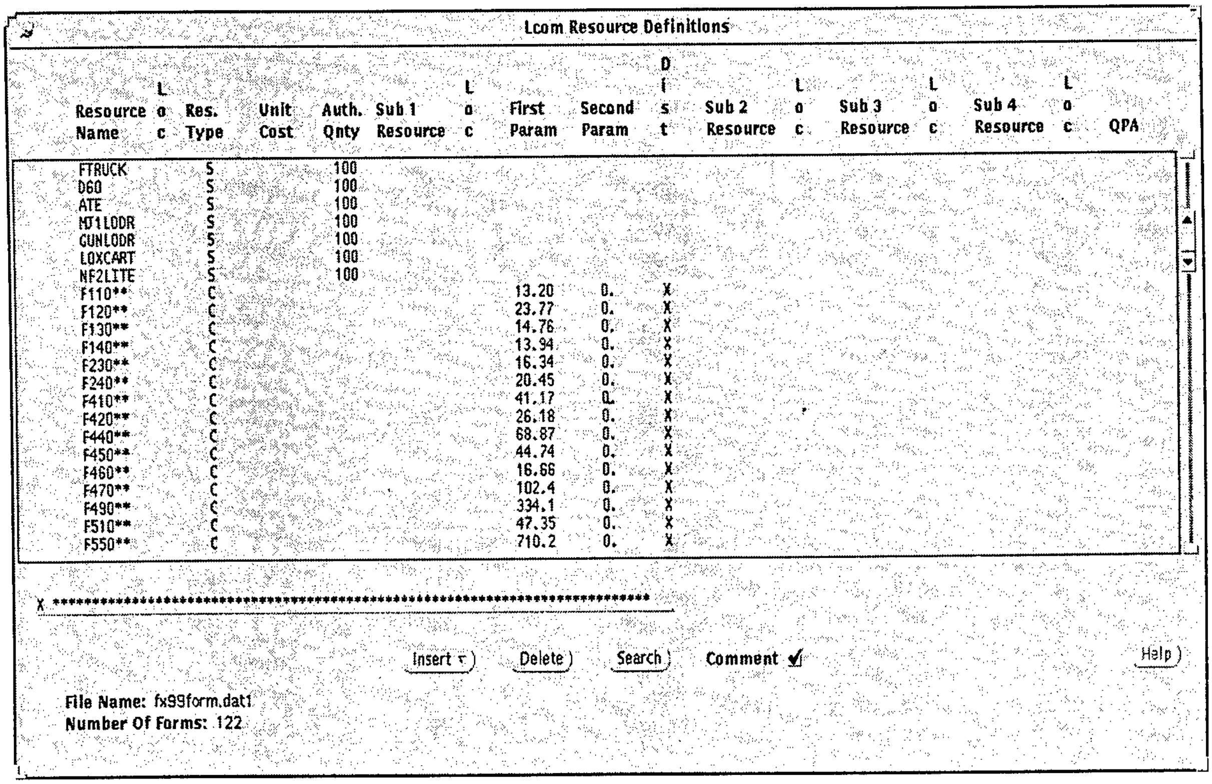


Figure A- 1.
Resource/Clock Forms Window

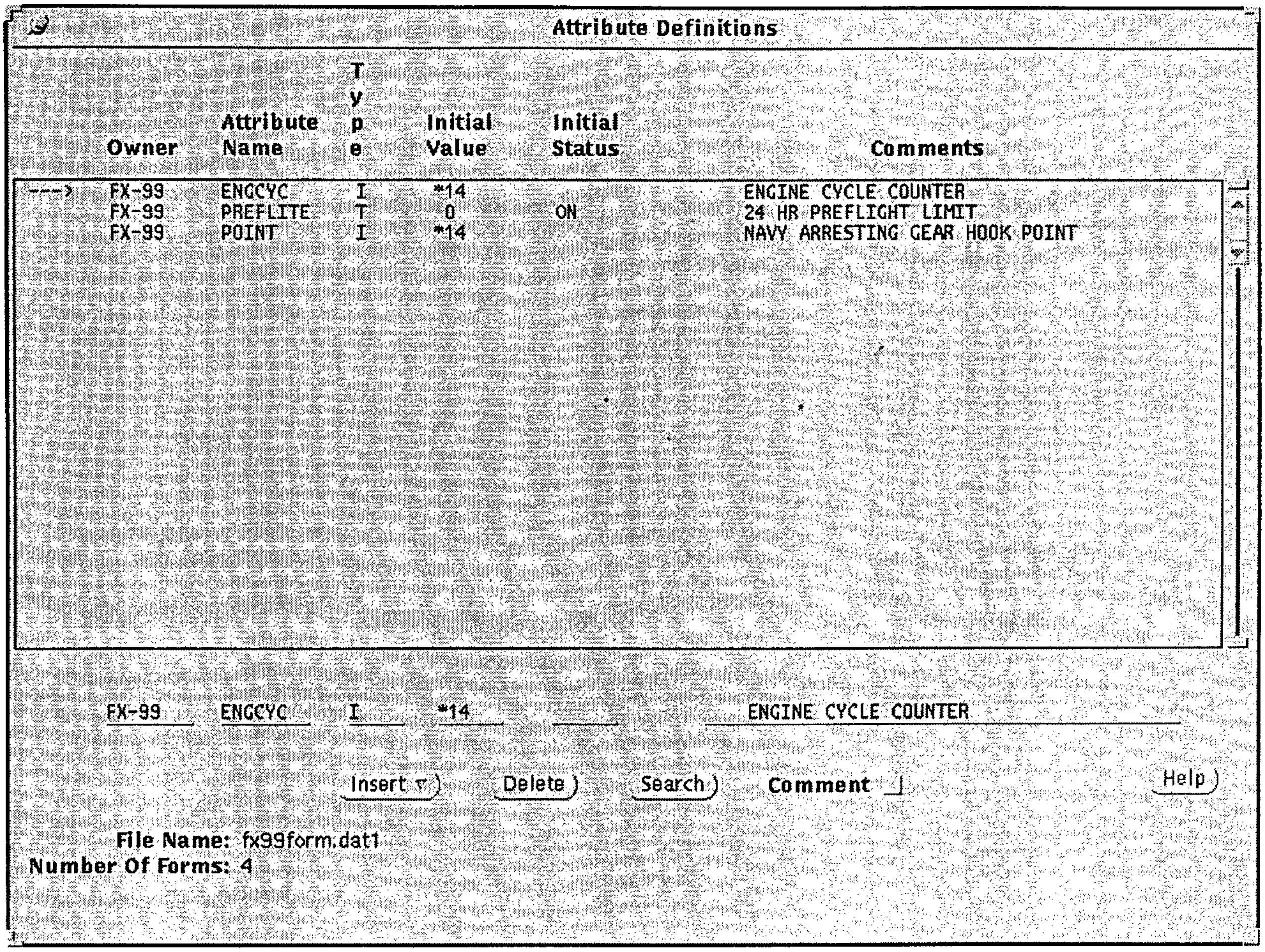
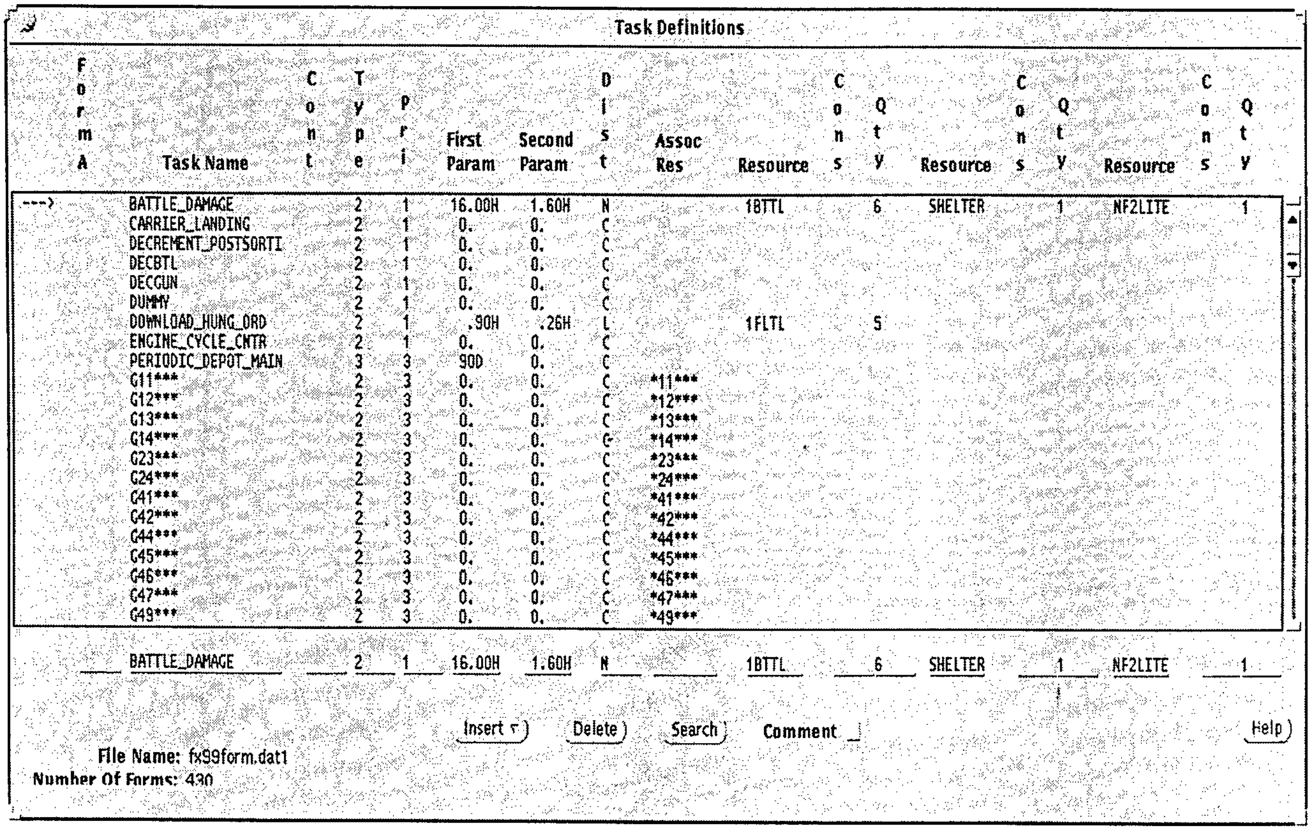


Figure A- 2.
Attribute Definitions



Task Definitions

Form A	Task Name	Cont	Type	Pri	First Param	Second Param	Dist	Assoc Res	Resource	Cons	Qty	Resource	Cons	Qty	Resource	Cons	Qty
--->	BATTLE_DAMAGE		2	1	16.00H	1.60H	N		1BTTL		6	SHELTER		1	NF2LITE		1
	CARRIER_LANDING		2	1	0.	0.	C										
	DECREMENT_POSTSORTI		2	1	0.	0.	C										
	DECBTL		2	1	0.	0.	C										
	DECGUN		2	1	0.	0.	C										
	DUMMY		2	1	0.	0.	C										
	DOWNLOAD_HUNG_ORD		2	1	.90H	.26H	L		1FLTL		5						
	ENGINE_CYCLE_CNTR		2	1	0.	0.	C										
	PERIODIC_DEPOT_MAIN		3	3	90D	0.	C										
	G11***		2	3	0.	0.	C	*11***									
	G12***		2	3	0.	0.	C	*12***									
	G13***		2	3	0.	0.	C	*13***									
	G14***		2	3	0.	0.	C	*14***									
	G23***		2	3	0.	0.	C	*23***									
	G24***		2	3	0.	0.	C	*24***									
	G41***		2	3	0.	0.	C	*41***									
	G42***		2	3	0.	0.	C	*42***									
	G44***		2	3	0.	0.	C	*44***									
	G45***		2	3	0.	0.	C	*45***									
	G46***		2	3	0.	0.	C	*46***									
	G47***		2	3	0.	0.	C	*47***									
	G49***		2	3	0.	0.	C	*49***									
	BATTLE_DAMAGE		2	1	16.00H	1.60H	N		1BTTL		6	SHELTER		1	NF2LITE		1

Insert | Delete | Search | Comment | Help

File Name: fx99form.dat1

Number Of Forms: 430

Figure A-3.
Task Definitions

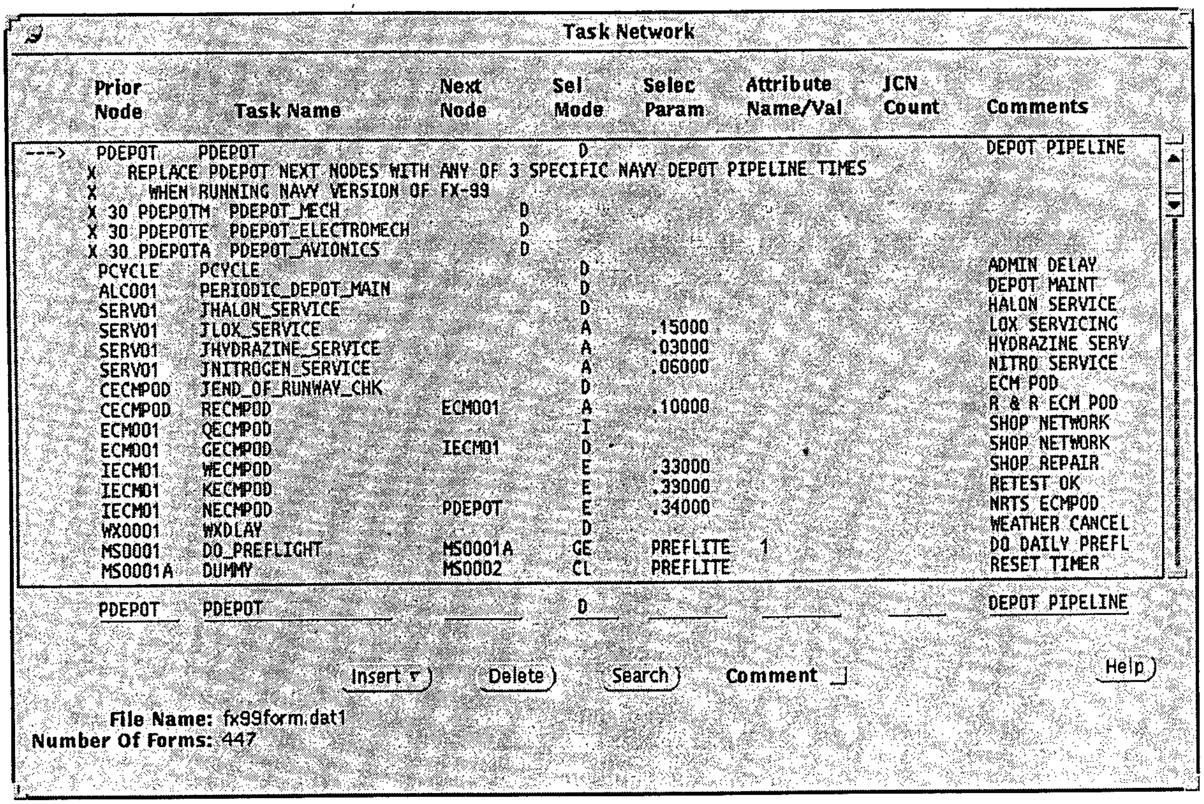


Figure A-4.
Task Network

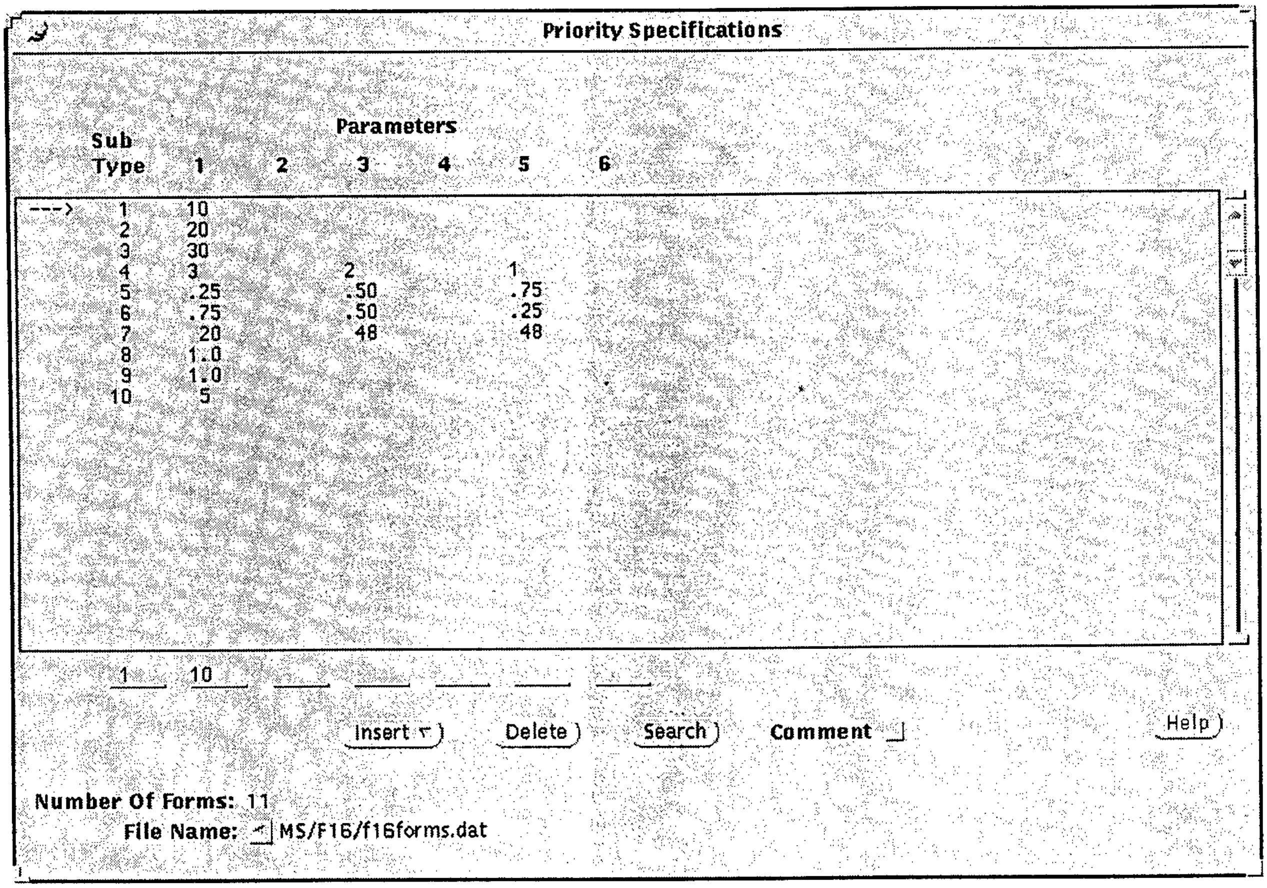


Figure A-5
Priority Specification

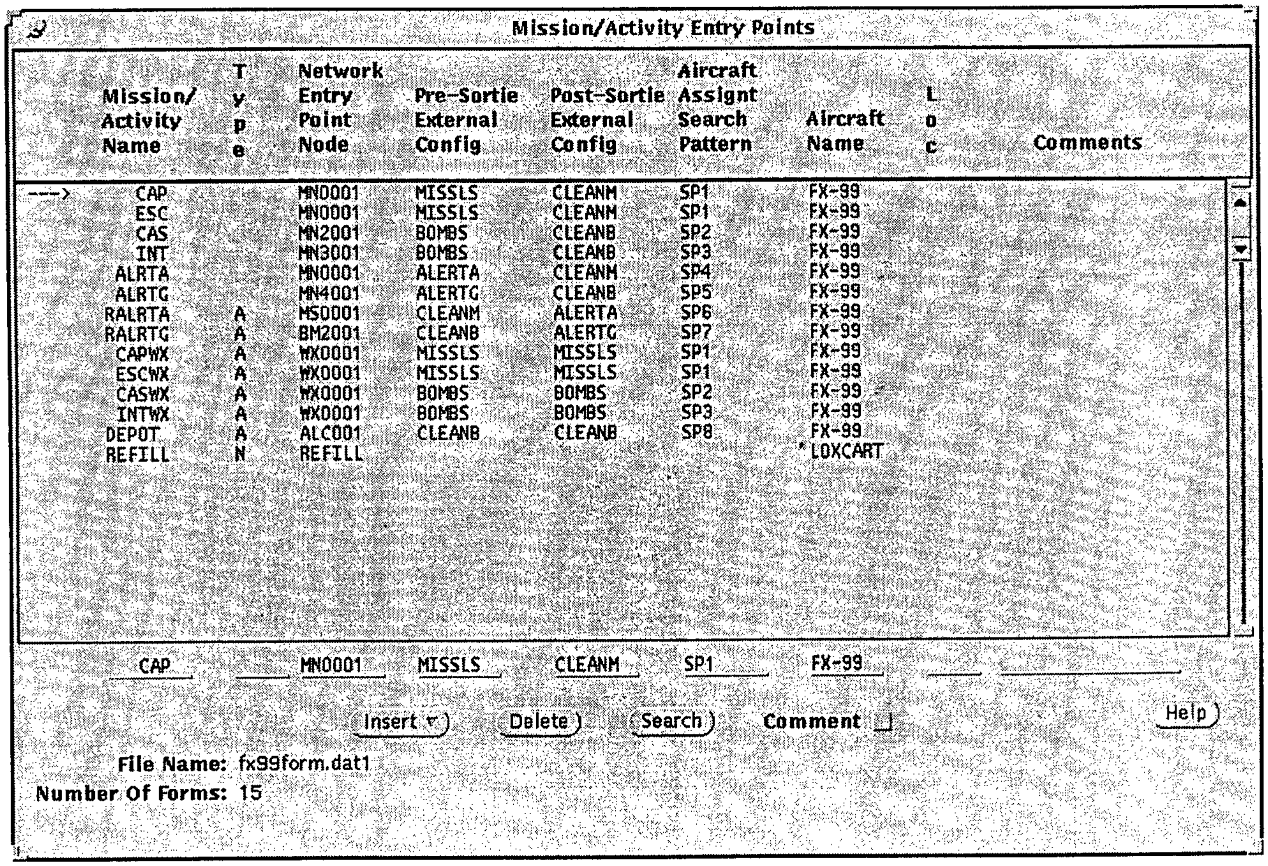
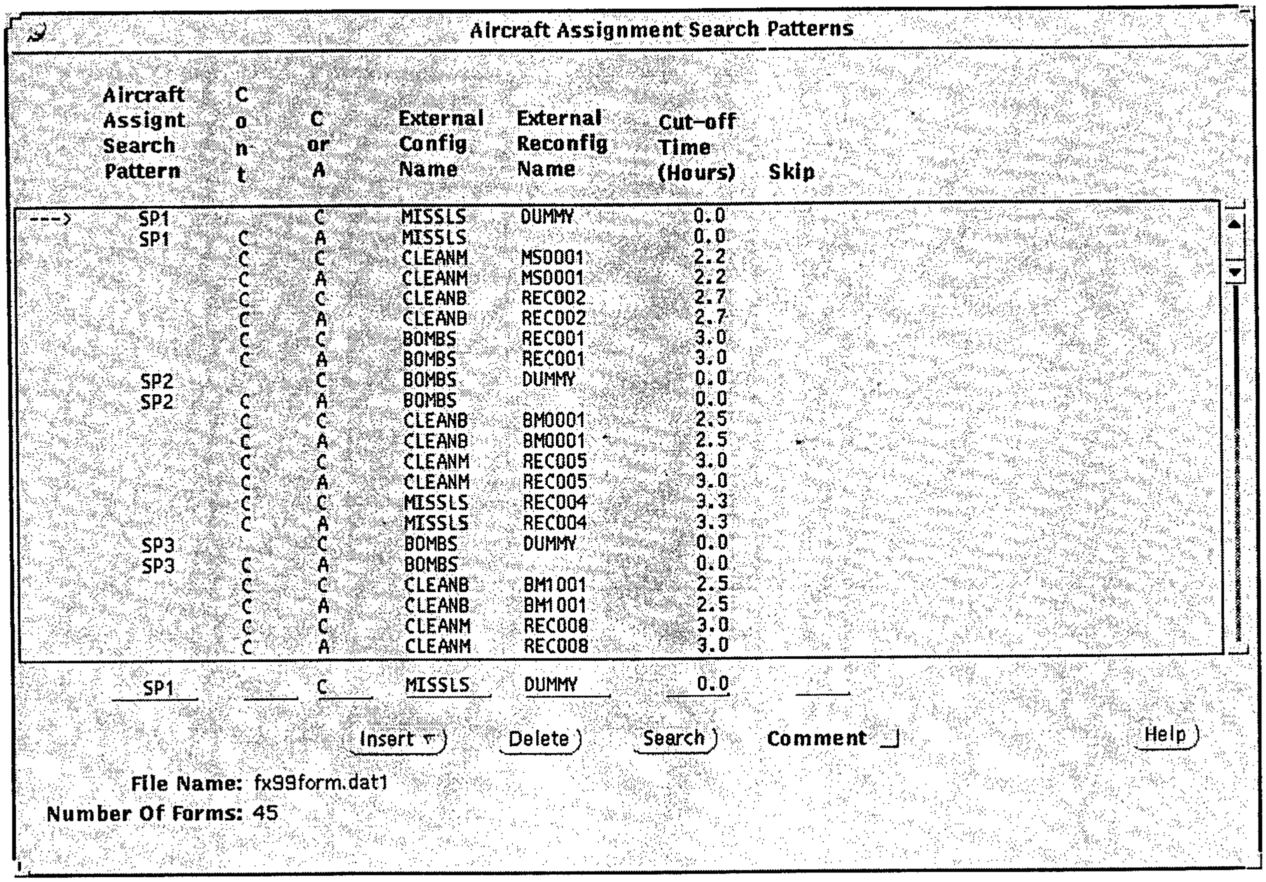


Figure A-6.
Mission /Activity Entry Points



Aircraft Assignment Search Patterns

	Aircraft Assignt Search Pattern	Cont	C or A	External Config Name	External Reconfig Name	Cut-off Time (Hours)	Skip
--->	SP1		C	MISSLS	DUMMY	0.0	
	SP1	C	A	MISSLS		0.0	
		C	C	CLEANM	MS0001	2.2	
		C	A	CLEANM	MS0001	2.2	
		C	C	CLEANB	REC002	2.7	
		C	A	CLEANB	REC002	2.7	
		C	C	BOMBS	REC001	3.0	
		C	A	BOMBS	REC001	3.0	
	SP2		C	BOMBS	DUMMY	0.0	
	SP2	C	A	BOMBS		0.0	
		C	C	CLEANB	BM0001	2.5	
		C	A	CLEANB	BM0001	2.5	
		C	C	CLEANM	REC005	3.0	
		C	A	CLEANM	REC005	3.0	
		C	C	MISSLS	REC004	3.3	
		C	A	MISSLS	REC004	3.3	
	SP3		C	BOMBS	DUMMY	0.0	
	SP3	C	A	BOMBS		0.0	
		C	C	CLEANB	BM1001	2.5	
		C	A	CLEANB	BM1001	2.5	
		C	C	CLEANM	REC008	3.0	
		C	A	CLEANM	REC008	3.0	

SP1 C MISSLS DUMMY 0.0

Insert Delete Search Comment Help

File Name: fx99form.dat1

Number Of Forms: 45

Figure A-7.
Aircraft Assignment Search Patterns

Sortie Generation Data

Day	Number Missions/ Activities	Takeoff/ Activity Time	Aircraft/ Resource Name	Loc	Mission/ Activity Name	Mission Size Min	Mission Size Max	Mission Size Spre	Mission Length First Param	Mission Length Second Param	Dist	Lead Time	Cancl Time	Pri	Cycle Int	Cycle Stop	Type	Taper
---> 1	10	*11	FX-99		CAP	02	02	00	2.200	.440	N	3.0	.5	2	1	990		
1	18	*11	FX-99		ESC	02	02	00	2.200	.440	N	3.0	.5	2	1	990		
1	47	*11	FX-99		CAS	02	02	00	1.700	.340	N	3.0	.5	2	1	990		
1	20	*12	FX-99		INT	01	01	00	1.700	.340	N	3.0	.5	2	1	990		
X NAVY PREPARES 2 SPARES FOR EACH MISSION																		
X75 1 10 *11 FX-99 CAP 0202 02 2.200 .440N 3.0 .52 1990																		
X75 1 10 *11 FX-99 ESC 0202 02 2.200 .440N 3.0 .52 1990																		
X75 1 47 *11 FX-99 CAS 0202 02 1.700 .340N 3.0 .52 1990																		
X75 1 20 *12 FX-99 INT 0101 02 1.700 .340N 3.0 .52 1990																		
1	6	*13	FX-99		ALRTA	02	02	00	.700	.140	N	.7	.2	1	1	990		
1	6	*11	FX-99		ALRTG	02	02	00	1.500	.300	N	1.5	.2	1	1	990		
1	6	*13	FX-99		RALRTA	02	02	00					12.0	1	1	990	3	
1	6	*11	FX-99		RALRTG	02	02	00					12.0	1	1	990	3	
1	1	0100	FX-99		DEPOT	01	01	00					6.0	1				
1	1	2200	LOXCART		REFILL	01	01	00						1	1	990		
30	1	1800	FX-99		CAPWX	02	02	00					2.7	2			2	
30	1	1800	FX-99		ESCWX	02	02	00					2.7	2			2	
30	1	1800	FX-99		CASWX	02	02	00					2.2	2			2	
30	1	1800	FX-99		INTWX	01	01	00					2.2	2			2	
1	10	*11	FX-99		CAP	02	02	00	2.200	.440	N	3.0	.5	2	1	990		

Insert Delete Search Comment Help

File Name: fx99form.dat1
Number Of Forms: 20

Figure A-8.
Sortie Generation Data

U.S. Government Printing Office 1996 750-071/40006